

**UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

CHAPTERHOUSE, LLC,

Plaintiff

v.

SHOPIFY, INC.,

Defendant

Case No. 2:18-cv-00300

JURY TRIAL DEMANDED

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Chapterhouse, LLC (“Chapterhouse”) hereby asserts the following claims for patent infringement against Defendant Shopify, Inc. (“Defendant” or “Shopify”), and alleges, on information and belief, as follows:

THE PARTIES

1. Chapterhouse is a limited liability company organized and existing under the laws of the Texas with its principal place of business at 17330 Preston Road, Suite 200, Dallas, Texas 75252.
2. Defendant is a Canadian company with a principal address at 150 Elgin Street 8th Floor, Ottawa, ON K2P 1L4, Canada.

JURISDICTION AND VENUE

3. This action arises under the patent laws of the United States, 35 U.S.C. § 1, *et seq.* This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).
4. On information and belief, the Court has personal jurisdiction over Defendant because Defendant has committed, and continues to commit, acts of infringement in the state of Texas, has

conducted business in the state of Texas, and/or has engaged in continuous and systematic activities in the state of Texas.

5. On information and belief, Defendant's instrumentalities that are alleged herein to infringe were and continue to be used, imported, offered for sale, and/or sold in the Eastern District of Texas.

6. Venue is proper in the Eastern District of Texas pursuant to 28 U.S.C. § 1391(c) because Defendant is not a resident in the United States. *In re HTC Corp.*, No. 2018-130, 2018 (Fed. Cir. May 9, 2018).

SHOPIFY

7. Upon information and belief, Defendant Shopify makes, uses, imports, sells, and/or offers for sale an ecommerce platform system ("System"). The System is described by the Shopify website (www.shopify.com) and is exemplified by the following references:

- "API Announcement: Shopify's Growing, and So are Our Integers! – Shopify Engineering" ("**API**"), available at <https://shopifyengineering.myshopify.com/blogs/archived/17489016-api-announcement-shopifys-growing-and-so-are-our-integers> (last accessed June 19, 2018);
- "Best Ecommerce Platform in India | Custom Ecommerce Software for Your Needs | Shopify" ("**ecommerce Platform**"), available at <https://www.shopify.in/?ref=easytravelr> (last accessed June 19, 2018);
- "Dribbble – shopify-pos-order.png by Nick Looijmans" ("**Dribbble**"), available at <https://dribbble.com/shots/1457266-Shopify-POS-Orders-redesign/attachments/216058> (last accessed June 19, 2018);
- "Exporting to a CSV file – Orders – Shopify Help Center" ("**Exporting**"), available at <https://help.shopify.com/manual/orders/export-orders> (last accessed June 19, 2018);
- "Get More Sales with These 5 Email Receipt Marketing Tactics" ("**Marketing Tactics**"), available at <https://www.shopify.com/blog/18058812-get-more-sales-with-these-5-email-receipt-marketing-tactics> (last accessed July 5, 2018);

- “New order notification – Notifications – Shopify Help Center” (“**New Order Notifications**”), available at <https://help.shopify.com/manual/sell-online/notifications/order-notifications> (last accessed June 19, 2018);
- “Notification variables reference – Notifications – Shopify Help Center” (“**Notifications**”), available at <https://help.shopify.com/manual/sell-online/notifications/email-variables> (last accessed June 19, 2018);
- “Order – Orders – Shopify Help Center” (“**Orders**”), available at <https://help.shopify.com/api/reference/orders/order> (last accessed June 19, 2018);
- “Orders – Using Shopify – Shopify Help Center” (“**Using Shopify**”), available at <https://help.shopify.com/manual/orders> (last accessed June 19, 2018);
- “Sales – Reports – Shopify Help Center” (“**Sales Reports**”), available at <https://help.shopify.com/manual/reports-and-analytics/shopify-reports/report-types/sales-report> (last accessed June 19, 2018);
- “Spently – Sell More with Custom Email Notifications” (“**Spently**”), available at <https://apps.shopify.com/spently> (last accessed July 5, 2018); and
- “Web Server Information” (“**Server Information**”) available at <http://browserspy.dk/webserver.php> (last accessed June 19, 2018).

COUNT I

(Infringement of U.S. Patent No. 7,552,087)

8. Plaintiff incorporates paragraphs 1-7 herein by reference.
9. Plaintiff is the owner, by assignment, of U.S. Patent No. 7,552,087 (the “’087 Patent”), entitled ELECTRONIC TRANSACTION RECEIPT SYSTEM AND METHOD, which issued on June 23, 2009. A copy of the ’087 Patent is attached as **Exhibit A**.
10. The ’087 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.
11. Upon information and belief, Defendant has infringed and continues to infringe one or more claims, including Claim 1, of the ’087 Patent by making, using, importing, selling, and/or, offering for sale the System covered by one or more claims of the ’087 Patent. Defendant has

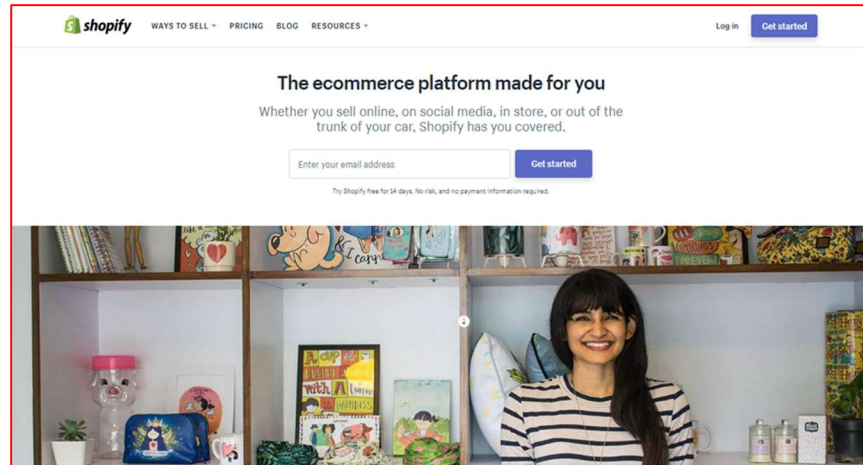
infringed and continues to infringe the '087 Patent either directly or through the acts of contributory infringement or inducement in violation of 35 U.S.C. § 271. Defendant has been on notice of the '087 Patent at least as early as the date it received service of this complaint.

12. Defendant sells, offers to sell, and/or uses the System, known as Shopify POS, and any similar products, which infringe at least Claim 1 of the '087 Patent.

13. Claim 1 of the '087 Patent recites:

1. A system for generating de-identified electronic receipts, comprising:
 - a. a database for storing a plurality of sales transactions between a plurality of buyers and a plurality of sellers, wherein each of the plurality of sales transactions includes an identity of one of the plurality of buyers;
 - b. an electronic receipt generator for:
 - i. generating an electronic receipt for each of the plurality of sales transactions, wherein each electronic receipt comprises detailed data describing the sales transaction, including sale price, total transaction amount, item description, and information indicating completion of the sales transaction; and
 - ii. generating de-identified data from each of the electronic receipts, wherein the de-identified data describes each of the plurality of sales transactions, but does not personally identify any one of the plurality of buyers.

14. The Shopify System is a system for generating de-identified electronic receipts. *See, e.g., Server Information*; and *Sales Reports*. An example is illustrated below:



ecommerce Platform.

Common sales report terms

The sales reports use the following common terms:

Orders	The number of orders that were placed on a given date.
Gross sales	Equates to product price x quantity (before taxes, shipping, discounts, and returns) for a collection of sales. Canceled, pending, and unpaid orders are included. Test and deleted orders are not included.
Discounts	<ul style="list-style-type: none"> Equates to line item discount + order level discount share for a collection of sales. The total dollar value reduction applied to a sale in the form of discounts on products, collections, or entire orders. Discounts that are applied to an entire order are proportionally applied to the sales for the order. All discounts are applied before taxes. Discounts are created using discount codes, not the compare at price.
Returns	The value of goods returned by a customer.
Net sales	Equates to gross sales - discounts - returns .
Shipping	Equates to shipping charges - shipping discounts - refunded shipping amounts .
Tax	The total amount of taxes based on the orders.
Total sales	<p>Equates to gross sales - discounts - returns + taxes + shipping charges.</p> <p>Total sales will be a positive number for a sale on the date that an order was placed, and a negative number for a return on the date that an order was returned.</p>

Sales Reports.

15. The Shopify System includes a database (e.g. MySQL) that stores a plurality of sales transactions (e.g. transaction history, order and product information) between a plurality of buyers and sellers. Each of these stored sales transactions includes an identity to one of the buyers. *See*, e.g., **API; Orders; and Using Shopify**. An example is illustrated below:

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "IPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "IPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

product_id: The ID for the product that the line item belongs to. Can be `null` if the original product associated with the order is deleted at a later date.

- **title:** The title of the product.
- **variant_id:** The ID of the product variant.
- **variant_title:** The title of the product variant.
- **vendor:** The name of the item's supplier.

Id.

An order is a customer's completed request to purchase one or more products from a shop. An order is created when a customer completes the checkout process, during which time they provide an email address or phone number, billing address and payment information.

Id.

customer	<pre>"customer": { "accepts_marketing": false, "created_at": "2012-03-13T16:09:55-04:00", "email": "bob.norman@hostmail.com", "phone": "+13125551212", "first_name": "Bob", "id": 207119551, "last_name": "Norman", "note": null, "orders_count": "0", "state": null, "total_spent": "0.00", "updated_at": "2012-03-13T16:09:55-04:00", "tags": "loyal" }</pre> <p>Information about the customer. The order might not have a customer and apps should not depend on the existence of a <code>customer</code> object. This value might be <code>null</code> if the order was created through Shopify POS. For more information about the <code>customer</code> object, see the Customer resource.</p>
----------	---

Id.

shopify

Search

Jonny B. Good

John's Apparel

Home

Orders

All orders

Drafts

Abandoned checkouts

Products

Customers

Analytics

Discounts

Apps

SALES CHANNELS

Online Store

Point of Sale

Facebook

Settings

Orders

Export

More actions

Create order

All

Open

Unfulfilled

Unpaid

Filter

Search orders

	Order	Date	Customer	Payment status	Fulfillment status	Total
<input type="checkbox"/>	#1029	1:40pm	John Smith	Pending	Unfulfilled	\$300.00
<input type="checkbox"/>	#1028	9:46am	John Smith	Pending	Partially Fulfilled	\$331.43
<input type="checkbox"/>	#1027	Mar 16, 12:52pm	George Splurman	Paid	Fulfilled	\$375.47
<input type="checkbox"/>	#1026	Mar 16, 12:51pm	—	Pending	Fulfilled	\$360.47
<input type="checkbox"/>	#1025	Mar 16, 12:17pm	—	Paid	Unfulfilled	\$143.82
<input type="checkbox"/>	#1024	Feb 9, 10:52am	—	Paid	Fulfilled	\$244.08

Using Shopify.

16. The Shopify System includes an electronic receipt generator. *See, e.g., Dribble.* An example is illustrated below:

Orders		Order Details	
February 14, 2014	11:22 AM #1-1032 \$442.03 Megan Gilbert	\$442.03 #1-1001 - Paid and Fulfilled Megan Gilbert (meggil@example.com) Sold by Devino Solano	
11:10 AM #1-1031	\$184.00 Kyle Brown	The Standard Sweatshirt The Standard Sweatshirt-Logo, M \$180.00	
10:45 AM #1-1030	\$256.21 Isaac Meivih	Shawl Neck Sweater Coat Shawl Neck Sweater Coat, M 2 x \$18.00 = \$36.00	
10:32 AM #1-1029	\$52.25 Connor Moth	The Wool Pullover The Wool Pullover, Charcoal Gray, M \$225.00	
February 13, 2014	10:16 PM #2654 \$89.12 Richard Medina	SUBTOTAL \$423.00 STATE TAX (4.5%) \$19.03	
5:10 PM #1-1028	\$69.21	GIFT CARD \$50.00 CODE: XXXX-XXXX-XXXX-R2D2	
4:56 PM #1-1027	\$574.12 Seth	MASTERCARD \$392.03	
4:21 PM #1-1026	\$74.58 George Salazar	Receipt Refund	
4:05 PM #1-1025	\$297.41 Ashleigh Stanton		

Dribble.

17. The System generates an electronic receipt for each of the plurality of sales transactions, wherein each electronic receipt comprises detailed data describing the sales transaction including the sales price, transaction amount, item description and information indicating completion of the sales transaction. *See, e.g., Dribble.* An example is illustrated below:

iPad 11:15 AM 93%		11:15 AM 4:14 PM	
Orders			
February 14, 2014			
11:22 AM #1-1032	\$442.03 Megan Gilbert	\$442.03 #1-1001 - Paid and Fulfilled	
11:10 AM #1-1031	\$184.00 Kyle Brown	Megan Gilbert (meggi@example.com) Sold by Devino Solano	
10:45 AM #1-1030	\$256.21 Isaac Melvin	The Standard Sweatshirt The Standard Sweatshirt-Logo, M \$180.00	
10:32 AM #1-1029	\$52.25 Connor Moth	Shawl Neck Sweater Coat Shawl Neck Sweater Coat, M 2 x \$18.00 = \$36.00	
February 13, 2014		The Wool Pullover The Wool Pullover, Charcoal Gray, M \$225.00	
10:16 PM #2654	\$89.12 Richard Medina	SUBTOTAL \$423.00	
5:10 PM #1-1028	\$69.21	STATE TAX (4.5%) \$19.03	
4:56 PM #1-1027	\$574.12 Seth	GIFT CARD \$50.00 CODE: XXXX-XXXX-XXXX-R2D2	
4:21 PM #1-1026	\$74.58 George Salazar	MASTERCARD \$392.03	
4:05 PM #1-1025	\$297.41 Ashleigh Stanton	Receipt Refund	

Dribble.

18. The System generates de-identified data from each of the electronic receipts. The de-identified data describes each of the sales transactions but does not personally identify any one of the buyers. *See, e.g., Sales Reports.* An example is illustrated below:

Common sales report terms

The sales reports use the following common terms:

Orders	The number of orders that were placed on a given date.
Gross sales	Equates to product price x quantity (before taxes, shipping, discounts, and returns) for a collection of sales. Canceled, pending, and unpaid orders are included. Test and deleted orders are not included.
Discounts	<ul style="list-style-type: none"> Equates to line item discount + order level discount share for a collection of sales. The total dollar value reduction applied to a sale in the form of discounts on products, collections, or entire orders. Discounts that are applied to an entire order are proportionally applied to the sales for the order. All discounts are applied before taxes. Discounts are created using discount codes, not the compare at price.
Returns	The value of goods returned by a customer.
Net sales	Equates to gross sales - discounts - returns .
Shipping	Equates to shipping charges - shipping discounts - refunded shipping amounts .
Tax	The total amount of taxes based on the orders.
Total sales	<p>Equates to gross sales - discounts - returns + taxes + shipping charges.</p> <p>Total sales will be a positive number for a sale on the date that an order was placed, and a negative number for a return on the date that an order was returned.</p>

Sales Reports.

19. Plaintiff has been damaged by Defendant's infringement of the '087 Patent.

COUNT II **(Infringement of U.S. Patent No. 7,742,989)**

20. Plaintiff incorporates paragraphs 1-7 herein by reference.

21. Plaintiff is the owner, by assignment, of U.S. Patent No. 7,742,989 (the "'989 Patent"), entitled DIGITAL RECEIPT GENERATION FROM INFORMATION ELECTRONICALLY READ FROM PRODUCT, which issued on June 22, 2010. A copy of the '989 Patent is attached as **Exhibit B**.

22. The '989 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

23. Upon information and belief, Defendant has infringed and continues to infringe one or more claims, including Claim 1, of the '989 Patent by making, using, importing, selling, and/or, offering for sale the System covered by one or more claims of the '989 Patent. Defendant has infringed and continues to infringe the '989 Patent either directly or through the acts of contributory infringement or inducement in violation of 35 U.S.C. § 271. Defendant has been on notice of the '989 Patent at least as early as the date it received service of this complaint.

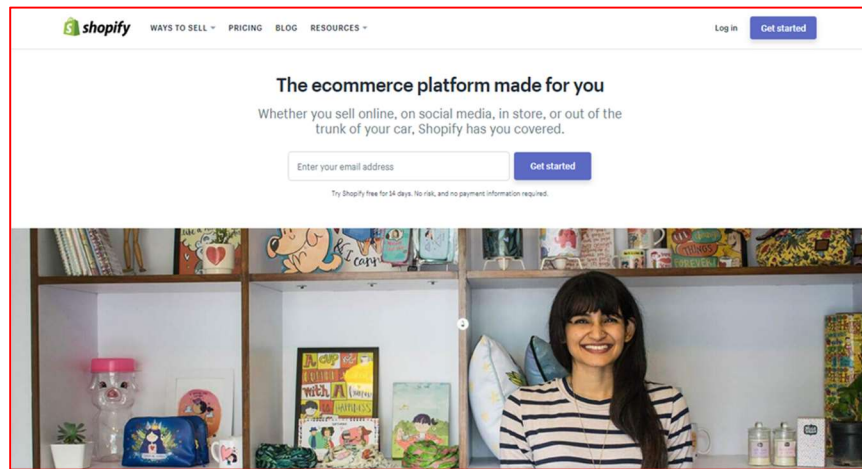
24. Defendant sells, offers to sell, and/or uses the System, known as Shopify POS, and any similar products, which infringe at least Claim 1 of the '989 Patent.

25. Claim 1 of the '989 Patent recites:

1. A system for collecting product information, comprising:
 - a. a database for storing a plurality of sales transactions between a plurality of buyers and a plurality of sellers, wherein each of the plurality of sales transactions is associated with a product that is purchased by a buyer of the plurality of buyers from a seller of the plurality of sellers, and includes product information associated with the product;
 - b. a device for collecting selected product information directly from a selected product, wherein the selected product is associated with a selected sales transaction; and
 - c. a processor for:
 - i. storing the selected product information and selected sales transaction in the database, as one of the plurality of sales transactions; and

- ii. transmitting the selected product information to the seller associated with the selected sales transaction.

26. The Shopify System is a system for collecting product information. *See, e.g., ecommerce Platform; and Notifications.* An example is illustrated below:



ecommerce Platform.

Property	Description
line.applied_discounts (POS and draft orders only)	List of discounts applied to this item (each discount has the title, code, amount, savings and type properties).
line.custom (Draft orders only)	(boolean) Returns true if the item is a custom line item for a draft order.
line.grams	Weight of a single item.
line.image	Returns the URL of the image associated with this line item. You can also use the img_url filter to get specific image sizes, for example {{ line.image img_url: 'small' }}
line.line_price	The price multiplied by the quantity for that item.
line.price	The price for a single item.
line.properties	Returns an array of custom information for an item. Line item properties are specified by the customer on the product page, before adding a product to the cart.
line.quantity	Quantity for that item.
line.requires_shipping	(boolean) Returns true if the variant for the item has the checkbox "This is a physical product" checked on the product page.
line.sku	SKU associated with the product.
line.taxable	(boolean) Returns true if the variant for the item has the checkbox 'Charge taxes on this product' checked on the product page.
line.title	The name of the product followed by a dash followed by the name of the variant. The variant name is not included when it is "Default Title".
line.variant.barcode	Barcode associated with the product.
line.variant.image	The image for the product variant. Only returns an image if there is a specific image assigned to the variant in the line item.
line.product.vendor	Vendor for the item.

Notifications.

27. The Shopify System includes a database (e.g. MySQL) for storing a plurality of sales transactions (e.g. transaction history, order, and product related information) between a plurality of buyers and a plurality of sellers, wherein each of the plurality of sales transactions is associated with a product that is purchased by a buyer of the plurality of buyers from a seller of the plurality of sellers, and includes product information associated with the product. *See*, e.g., **API; Orders; and Using Shopify**. An example is illustrated below:

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "iPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "iPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

customer	<pre>"customer": { "accepts_marketing": false, "created_at": "2012-03-13T16:09:55-04:00", "email": "bob.norman@hotmail.com", "phone": "+13125551212", "first_name": "Bob", "id": 207119551, "last_name": "Norman", "note": null, "orders_count": "0", "state": null, "total_spent": "0.00", "updated_at": "2012-03-13T16:09:55-04:00", "tags": "loyal" }</pre> <p>Information about the customer. The order might not have a customer and apps should not depend on the existence of a <code>customer</code> object. This value might be <code>null</code> if the order was created through Shopify POS. For more information about the <code>customer</code> object, see the Customer resource.</p>
----------	--

Id.

- **product_id:** The ID for the product that the line item belongs to. Can be `null` if the original product associated with the order is deleted at a later date.
- **quantity:** The number of items that were purchased.
- **requires_shipping:** Whether the item requires shipping.
- **sku:** The item's SKU (stock keeping unit).
- **title:** The title of the product.
- **variant_id:** The ID of the product [variant](#).
- **variant_title:** The title of the product variant.
- **vendor:** The name of the item's supplier.
- **name:** The name of the product variant.

Id.

Order	Date	Customer	Payment status	Fulfillment status	Total
#1028	1:40pm	John Smith	Pending	Unfulfilled	\$300.00
#1028	9:46am	John Smith	Pending	Partially Fulfilled	\$331.43
#1027	Mar 16, 12:52pm	George Spurlman	Paid	Fulfilled	\$375.47
#1026	Mar 16, 12:51pm	—	Pending	Fulfilled	\$360.47
#1025	Mar 16, 12:17pm	—	Paid	Unfulfilled	\$143.82
#1024	Feb 9, 10:52am	—	Paid	Fulfilled	\$244.08

Using Shopify.

28. The Shopify System includes a server for collecting selected product information directly from a selected product, wherein the selected product is associated with a selected sales transaction. *See, e.g., Server Information; and Orders.* An example is illustrated below:

Web Server Information	
Find out which web server is running a specific site. See information like web dedicated server ^{id} name, operating system, available modules, etc.	
Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com ^{id}

Server Information.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "iPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "iPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

29. The Shopify System includes a server that has a processor for storing the selected product information and selected sales transaction in the database, as one of the plurality of sales transactions (e.g. order). *See*, e.g., **Server Information; Orders**; and **Dribble**. An example is illustrated below:

Web Server Information	
Find out which web server is running a specific site. See information like web dedicated server id name, operating system, available modules, etc.	
Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com id

Server Information.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-W", "title": "IPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "IPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

iPad Wi-Fi

11:15 AM

Orders

4:14 PM

February 14, 2014

11:22 AM
#1-1032

\$442.03
Megan Gilbert

11:10 AM
#1-1031

\$184.00
Kyle Brown

10:45 AM
#1-1030

\$256.21
Isaac Melvin

10:32 AM
#1-1029

\$52.25
Connor Moth

February 13, 2014

10:16 PM
#2654

\$89.12
Richard Medina

5:10 PM
#1-1028

\$69.21

4:56 PM
#1-1027

\$574.12
Seth

4:21 PM
#1-1026

\$74.58
George Salazar


4:05 PM
#1-1025

\$297.41
Ashleigh Stanton


\$442.03

#1-1001 - Paid and Fulfilled


Megan Gilbert (meggil@example.com)
Sold by Devino Solano

The Standard Sweatshirt
The Standard Sweatshirt-Logo, M

\$180.00

Shawl Neck Sweater Coat
Shawl Neck Sweater Coat, M

2 x \$18.00 = \$36.00

The Wool Pullover
The Wool Pullover, Charcoal Gray, M


\$225.00

SUBTOTAL


\$423.00

STATE TAX (4.5%)

\$19.03

GIFT CARD
CODE: XXXX-XXXX-XXXX-R2D2

\$50.00

MASTERCARD



\$392.03

Receipt

Refund

Dribbble.

30. The Shopify System includes a server that has a processor for transmitting the selected product information to the seller associated with the selected sales transaction (e.g. a notification when a new order is placed). *See*, e.g., **Server Information**; **New Order Notifications**; and **Notifications**. An example is illustrated below:

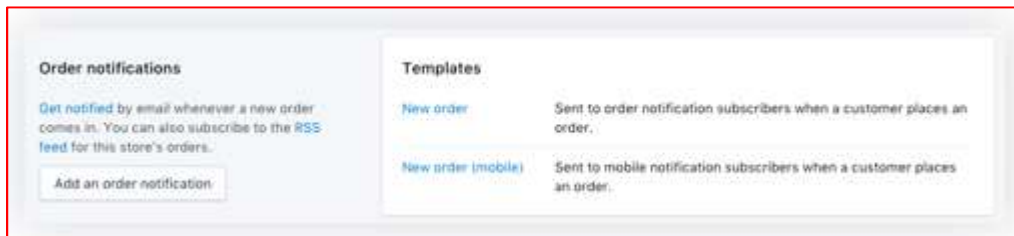
Web Server Information	
Find out which web server is running a specific site. See information like web dedicated server  name, operating system, available modules, etc.	
Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com 

Server Information.

New order notifications

You can create and edit email notifications to let you or a staff member know when a customer places an order. You need the **Orders** permission on your account to make changes to order notifications. [Learn more about staff permissions.](#)

New Order Notifications.



Id.

Property	Description
----------	-------------

Notifications.

transaction	Notifications have access to every transaction property. Please see our Transaction API documentation here for a full list of the properties.
-------------	---

Id.

<code>line.title</code>	The name of the product followed by a dash followed by the name of the variant. The variant name is not included when it is "Default Title".
<code>line.variant.barcode</code>	Barcode associated with the product.
<code>line.variant.image</code>	The image for the product variant. Only returns an image if there is a specific image assigned to the variant in the line item.
<code>line.product.vendor</code>	Vendor for the item.

Id.

31. Plaintiff has been damaged by Defendant's infringement of the '989 Patent.

32. COUNT III
(Infringement of U.S. Patent No. 8,112,356)

33. Plaintiff incorporates paragraphs 1-7 herein by reference.

34. Plaintiff is the owner, by assignment, of U.S. Patent No. 8,112,356 (the "'356 Patent"), entitled SYSTEM AND METHOD FOR PROVIDING AUTOMATED SECONDARY PURCHASE OPPORTUNITIES TO CONSUMERS, which issued on February 7, 2012. A copy of the '356 Patent is attached as **Exhibit C**.

35. The '356 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

36. Upon information and belief, Defendant has infringed and continues to infringe one or more claims, including Claim 1, of the '356 Patent by making, using, importing, selling, and/or, offering for sale the System covered by one or more claims of the '356 Patent. Defendant has infringed and continues to infringe the '356 Patent either directly or through the acts of contributory infringement or inducement in violation of 35 U.S.C. § 271. Defendant has been on notice of the '356 Patent at least as early as the date it received service of this complaint.

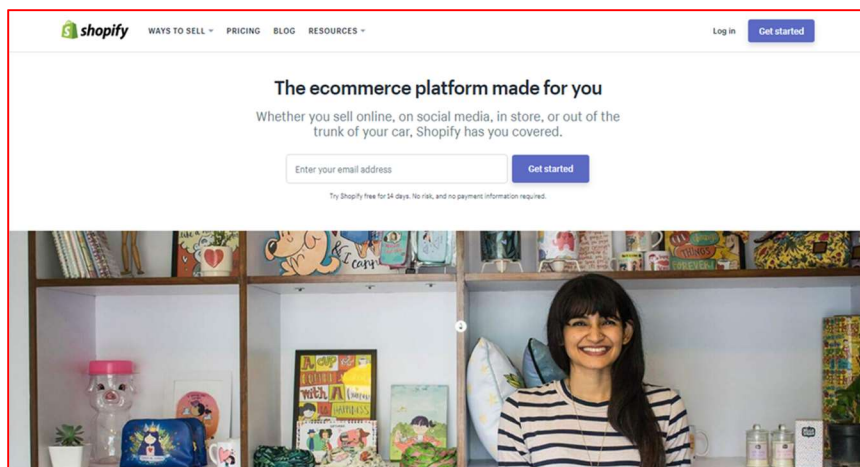
37. Defendant sells, offers to sell, and/or uses the System, known as Shopify POS, and any similar products, which infringe at least Claim 1 of the '356 Patent.

38. Claim 1 of the '087 Patent recites:

1. A system for providing an automated secondary purchase opportunity to a user, comprising:

- a. a database for storing a plurality of primary sales transactions between a plurality of buyers and a plurality of sellers;
- b. an electronic receipt generator for:
 - i. generating an electronic receipt for a selected one of the plurality of primary sales transactions associated with the user, wherein the electronic receipt comprises data describing the selected primary sales transaction, and the electronic receipt further includes a hyperlink associated with an inducement for a secondary purchase, the secondary purchase being commercially related to the selected primary sales transaction;
 - ii. transmitting the electronic receipt to a computing device operated by the user;
 - iii. receiving from the computer device a signal that the user has activated the hyperlink within the electronic receipt; and
 - iv. in response to the activation of the hyperlink, initiating a sales transaction relating to the secondary purchase.

39. The Shopify System is an eCommerce system for providing an automated secondary purchase opportunity to a user. *See, e.g., ecommerce Platform.* An example is illustrated below:



ecommerce Platform.

40. The Shopify System employs a database (e.g. MySQL) to store a plurality of sales transactions (e.g. transaction history, order and product information) between a plurality of buyers and sellers. *See*, e.g., **API; Orders; and Using Shopify**. An example is illustrated below:

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "IPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "IPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

product_id: The ID for the product that the line item belongs to. Can be `null` if the original product associated with the order is deleted at a later date.

- **title:** The title of the product.
- **variant_id:** The ID of the product variant.
- **variant_title:** The title of the product variant.
- **vendor:** The name of the item's supplier.

Id.

41. The Shopify System includes an electronic receipt generator. *See, e.g., Dribble.* An example is illustrated below:

Orders		11:15 AM	93%
February 14, 2014		4:14 PM	
11:22 AM #1-1032	\$442.03 Megan Gilbert	\$442.03 #1-1001 - Paid and Fulfilled	
11:10 AM #1-1031	\$184.00 Kyle Brown	Megan Gilbert (meggil@example.com) Sold by Devino Solano	
10:45 AM #1-1030	\$256.21 Isaac Melvin	The Standard Sweatshirt The Standard Sweatshirt-Logo, M	\$180.00
10:32 AM #1-1029	\$52.25 Connor Moth	Shawl Neck Sweater Coat Shawl Neck Sweater Coat, M	2 x \$18.00 = \$36.00
February 13, 2014		The Wool Pullover The Wool Pullover, Charcoal Gray, M	\$225.00
10:16 PM #2654	\$89.12 Richard Medina	SUBTOTAL	\$423.00
5:10 PM #1-1028	\$69.21	STATE TAX (4.5%)	\$19.03
4:56 PM #1-1027	\$574.12 Seth	GIFT CARD CODE: XXXX-XXXX-XXXX-R2D2	\$50.00
4:21 PM #1-1026	\$74.58 George Salazar	MASTERCARD	\$392.03
4:05 PM #1-1025	\$297.41 Ashleigh Stanton	Receipt Refund	

Dribble.

42. The System is operable to generate an electronic receipt for each of the plurality of sales transactions, wherein each electronic receipt comprises detailed data describing the sales transaction including the sales price, transaction amount, item description and information indicating completion of the sales transaction. *See, e.g., Dribble.* An example is illustrated below:

Orders		11:15 AM	93%
February 14, 2014		4:14 PM	
11:22 AM #1-1032	\$442.03 Megan Gilbert	\$442.03 #1-1001 - Paid and Fulfilled	
11:10 AM #1-1031	\$184.00 Kyle Brown	Megan Gilbert (meggil@example.com) Sold by Devino Solano	
10:45 AM #1-1030	\$256.21 Isaac Melvin	The Standard Sweatshirt The Standard Sweatshirt-Logo, M	\$180.00
10:32 AM #1-1029	\$52.25 Connor Moth	Shawl Neck Sweater Coat Shawl Neck Sweater Coat, M	2 x \$18.00 = \$36.00
February 13, 2014		The Wool Pullover The Wool Pullover, Charcoal Gray, M	\$225.00
10:16 PM #2654	\$89.12 Richard Medina	SUBTOTAL	\$423.00
5:10 PM #1-1028	\$69.21	STATE TAX (4.5%)	\$19.03
4:56 PM #1-1027	\$574.12 Seth	GIFT CARD CODE: XXXX-XXXX-XXXX-R2D2	\$50.00
4:21 PM #1-1026	\$74.58 George Salazar	MASTERCARD	\$392.03
4:05 PM #1-1025	\$297.41 Ashleigh Stanton	Receipt Refund	

43. Dribble.

44. With the integrated Spently app, the electronic receipt may further comprise a hyperlink associated with an inducement for a secondary purchase. The secondary purchase is commercially related to the selected primary sales transaction. *See, e.g., Marketing Tactics; and Spently.* An example is illustrated below:


Additional resources:

Read more about [upselling products to increase sales](#)

Try using an app like [Spently](#) to Create beautiful Order Confirmations, POS e-Receipts, Abandoned Cart Emails, Shipping Confirmations, Email Campaigns + more!

Learn more about Shopify's [email notification templates](#)

Marketing Tactics.



Supercharged Store Emails by Spently

Developed by Spently

★★★★★ 300 reviews

YouTube

Get

Price	From \$0.00 / month
Starter, 0.00\$/month	<ul style="list-style-type: none"> -All 15 email notifications, plus template themes -Recommended products, unique discount codes, and referral links -Store & Customer Analytics -Email Support -Up to 10 orders/month
Basic, 19.99\$/month	<p>Everything in Starter, plus:</p> <ul style="list-style-type: none"> -Automated follow-ups (after orders, shipments, and abandoned checkouts) -Up to 50 orders/month -Spently Watermark Removal
Pro, 39.99\$/month	<p>Everything in Basic, plus:</p> <ul style="list-style-type: none"> -Priority Support -Additional Marketing Components -Up to 100 orders/month

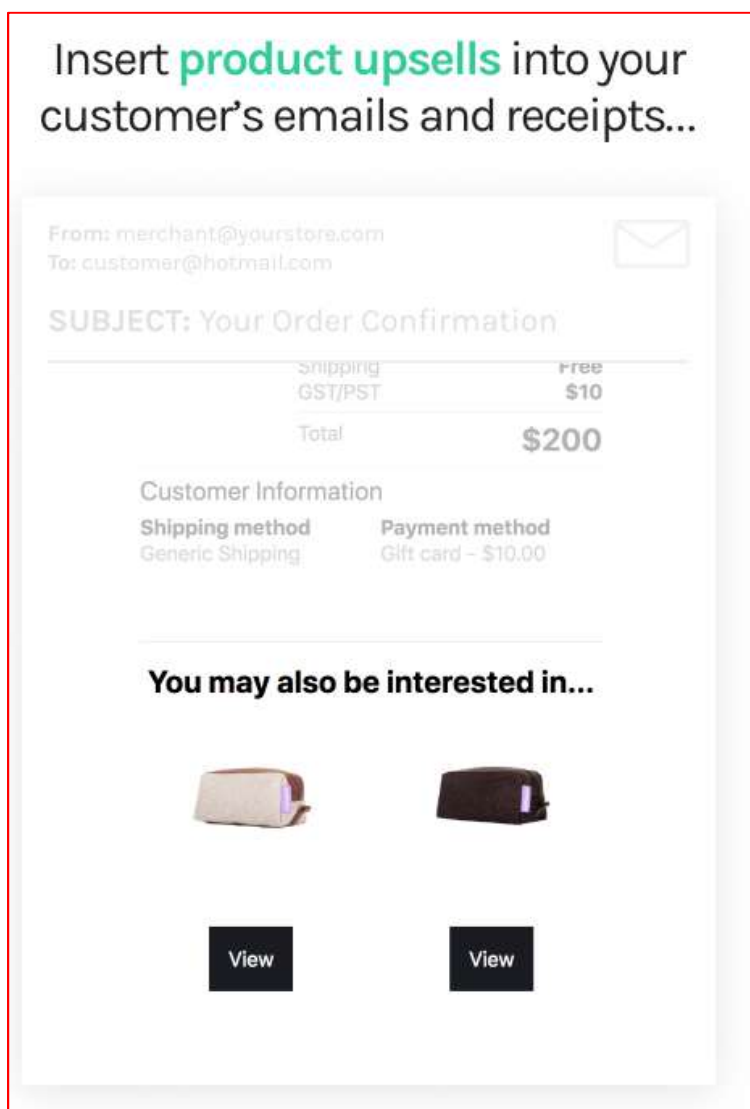
Spently.

2. Spently allows you to engage with your customers using marketing components that are embedded right within your Shopify notification emails — these emails have **substantially higher open rate paired with greater revenue generated per email**, as opposed to newsletters.

a) Drive sales through awareness using **Product Recommendations**: Spently can insert smart product upsells based on your customer's past purchasing behavior right within your notification emails.

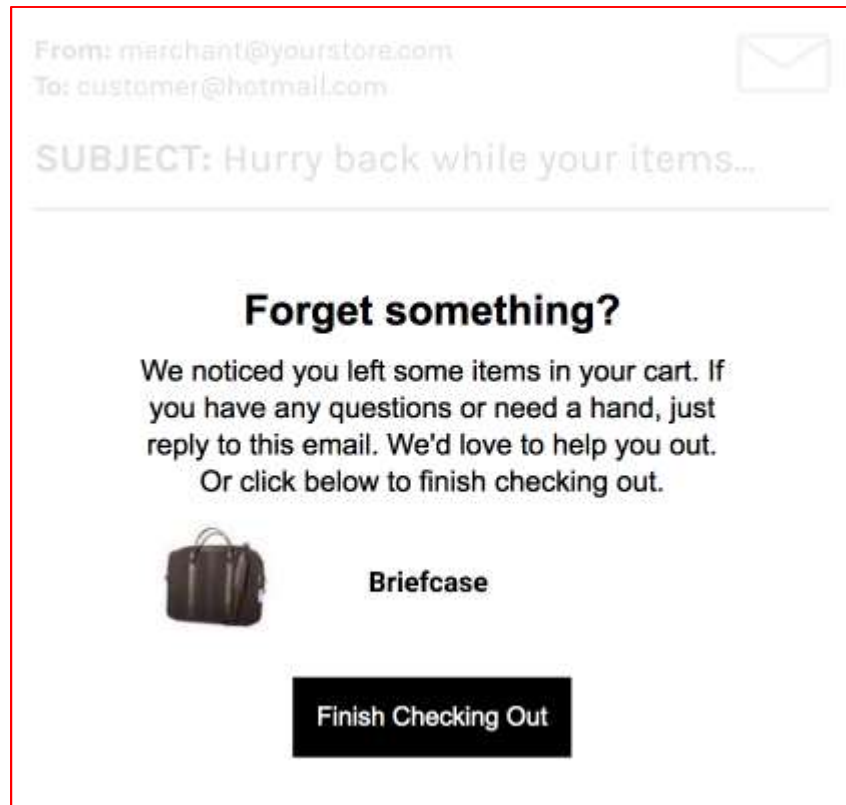
Id.

45. The Shopify System is operable to transmit the electronic receipt to a computing device operated by the user. *See, e.g., Spently*. An example is illustrated below:



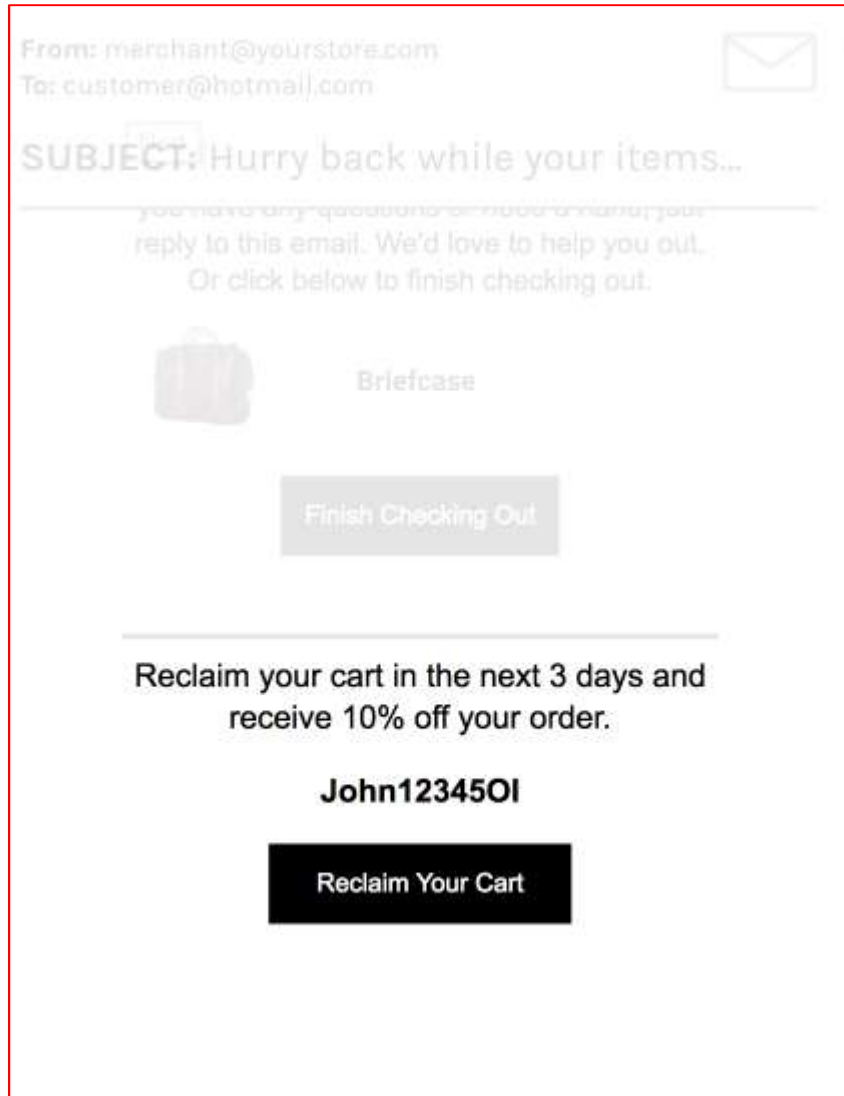
Spently.

46. The Shopify System is operable to receive a signal that the user has activated the hyperlink within the electronic receipt. *See, e.g., Spently.* An example is illustrated below:



Spently.

47. In response to the activation of the hyperlink, Shopify is operable to initiate a sales transaction relating to the secondary purchase. *See, e.g., Spently.* An example is illustrated below:



Spently.

48. Plaintiff has been damaged by Defendant's infringement of the '356 Patent.

COUNT IV
(Infringement of U.S. Patent No. 8,606,698)

49. Plaintiff incorporates paragraphs 1-7 herein by reference.

50. Plaintiff is the owner, by assignment, of U.S. Patent No. 8,606,698 (the "'698 Patent"), entitled ELECTRONIC TRANSACTION RECEIPT SYSTEM AND METHOD, which issued on December 10, 2013. A copy of the '698 Patent is attached as **Exhibit D**.

51. The '698 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

52. Upon information and belief, Defendant has infringed and continues to infringe one or more claims, including Claim 1, of the '698 Patent by making, using, importing, selling, and/or, offering for sale the System covered by one or more claims of the '698 Patent. Defendant has infringed and continues to infringe the '698 Patent either directly or through the acts of contributory infringement or inducement in violation of 35 U.S.C. § 271. Defendant has been on notice of the '698 Patent at least as early as the date it received service of this complaint.

53. Defendant sells, offers to sell, and/or uses the System, known as Shopify POS, and any similar products, which infringe at least Claim 1 of the '698 Patent.

54. Claim 1 of the '698 Patent recites:

1. A system for storing transaction receipts associated with a plurality of sellers and a plurality of buyers, comprising:

a database;

a device for:

receiving from a particular seller of the plurality of sellers a transaction receipt associated with a transaction that has occurred between a particular buyer of the plurality of buyers and the particular seller;

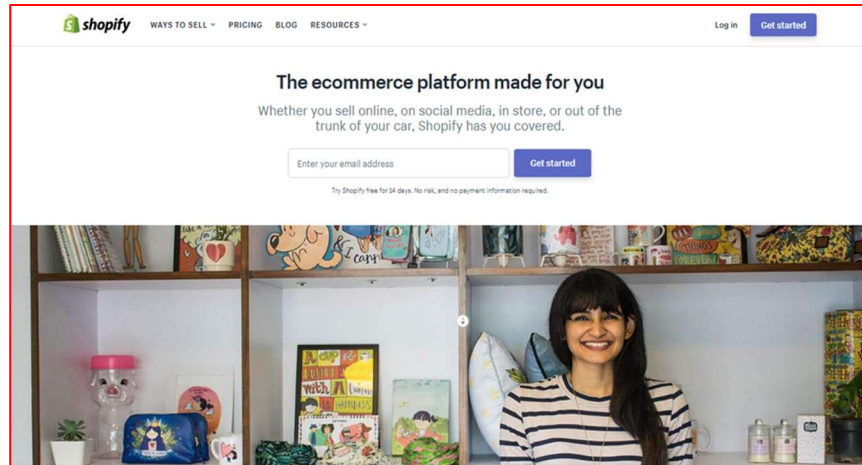
storing the transaction receipt in the database as one of a plurality of transaction receipts between the plurality of buyers and the plurality of sellers;

receiving a request from a requester to transmit a selected one of the plurality of transaction receipts; and

transmitting the selected one of the plurality of transaction receipts to the requester.

55. The Shopify System is a system for storing transaction receipts associated with a plurality of sellers and a plurality of buyers. *See, e.g.,* **ecommerce Platform**; and **API**.

56. An example is illustrated below:



ecommerce Platform.

57. The Shopify System includes a database (e.g. MySQL). *See, e.g.,* **API**. An example is illustrated below:

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

58. The Shopify System includes a device (e.g. server) for receiving from a particular seller of the plurality of sellers a transaction receipt associated with a transaction that has occurred between a particular buyer of the plurality of buyers and the particular seller. *See, e.g.,* **Server Information**; **Using Shopify**; and **Orders**. An example is illustrated below:

Web Server Information

Find out which web server is running a specific site. See information like web [dedicated server](#) ^{id} name, operating system, available modules, etc.

Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com ^{id}

Server Information.

Order	Date	Customer	Payment status	Fulfillment status	Total
#1029	1:40pm	John Smith	Pending	Unfulfilled	\$300.00
#1028	9:46am	John Smith	Pending	Partially Fulfilled	\$331.43
#1027	Mar 16, 12:52pm	George Spurman	Paid	Fulfilled	\$375.47
#1026	Mar 16, 12:51pm	—	Pending	Fulfilled	\$360.47
#1025	Mar 16, 12:17pm	—	Paid	Unfulfilled	\$143.82
#1024	Feb 9, 10:52am	—	Paid	Fulfilled	\$244.08

Using Shopify.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "IPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "IPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.

59. The Shopify System includes a device (e.g. server) for storing the transaction receipt in the database as one of a plurality of transaction receipts between the plurality of buyers and the plurality of sellers (e.g. by issuing unique IDs corresponding to individual transactions). *See, e.g., Server Information; and API.* An example is illustrated below:

Web Server Information	
Find out which web server is running a specific site. See information like web dedicated server ^{id} name, operating system, available modules, etc.	
Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com ^{id}

Server Information.

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

60. The Shopify System includes a device (e.g. server) for receiving a request from a requester to transmit a selected one of the plurality of transaction receipts. *See*, e.g., **Server Information; API; Orders;** and **Dribble**. An example is illustrated below:

Web Server Information

Find out which web server is running a specific site. See information like web [dedicated server](#) ^{id} name, operating system, available modules, etc.

Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com ^{id}

Server Information.

As Shopify and our merchants have grown, the sheer amount of data we deal with has grown as well. With that in mind, we've been forced to transition towards ID columns (and ID references) from 32-bit to 64-bit to ensure that we can continue to issue unique IDs.

MySQL, Ruby on Rails, and most other technologies default ID columns to INT(11). While we're very excited that we're hitting this limitation (problems caused by massive growth are the best problems to have), it is going to require Shopify App and integration developers to make the same transition to ensure compatibility with Shopify in the future.

API.

line_items	<pre> "line_items": [{ "fulfillable_quantity": 1, "fulfillment_service": "amazon", "fulfillment_status": "fulfilled", "grams": 500, "id": 669751112, "price": "199.99", "product_id": 7513594, "quantity": 1, "requires_shipping": true, "sku": "IPOD-342-N", "title": "iPod Nano", "variant_id": 4264112, "variant_title": "Pink", "vendor": "Apple", "name": "iPod Nano - Pink", "gift_card": false, "properties": [{ "name": "custom engraving", "value": "Happy Birthday Mom!" }], "taxable": true, "tax_lines": [{ "title": "State Tax", "price": "3.98", </pre>
------------	---

Orders.



- **product_id:** The ID for the product that the line item belongs to. Can be `null` if the original product associated with the order is deleted at a later date.
- **quantity:** The number of items that were purchased.
- **requires_shipping:** Whether the item requires shipping.
- **sku:** The item's SKU (stock keeping unit).
- **title:** The title of the product.
- **variant_id:** The ID of the product variant.
- **variant_title:** The title of the product variant.
- **vendor:** The name of the item's supplier.
- **name:** The name of the product variant.

Id.

iPad 11:15 AM 93%		11:15 AM 4:14 PM	
Orders			
February 14, 2014			
11:22 AM #1-1032	\$442.03 Megan Gilbert	\$442.03 #1-1001 - Paid and Fulfilled	
11:10 AM #1-1031	\$184.00 Kyle Brown	Megan Gilbert (meggli@example.com) Sold by Devino Solano	
10:45 AM #1-1030	\$256.21 Isaac Melvin	The Standard Sweatshirt The Standard Sweatshirt-Logo, M \$180.00	
10:32 AM #1-1029	\$52.25 Connor Moth	Shawl Neck Sweater Coat Shawl Neck Sweater Coat, M 2 x \$18.00 = \$36.00	
February 13, 2014		The Wool Pullover The Wool Pullover, Charcoal Gray, M \$225.00	
10:16 PM #2654	\$89.12 Richard Medina	SUBTOTAL \$423.00	
5:10 PM #1-1028	\$69.21	STATE TAX (4.5%) \$19.03	
4:56 PM #1-1027	\$574.12 Seth	GIFT CARD \$50.00 CODE: XXXX-XXXX-XXXX-R2D2	
4:21 PM #1-1026	\$74.58 George Salazar	MASTERCARD \$392.03	
4:05 PM #1-1025	\$297.41 Ashleigh Stanton	Receipt Refund	

Dribbble.

61. The Shopify System includes a device (e.g. server) for transmitting the selected one of the plurality of transaction receipts to the requester. *See, e.g., Server Information; and Exporting.* An example is illustrated below:

Web Server Information	
Find out which web server is running a specific site. See information like web dedicated server  name, operating system, available modules, etc.	
Header	Value
Web server	nginx
HTTP response code	200
Server Address	http://www.shopify.com 

Server Information.

Export orders

You can export orders along with their transaction histories or you can export only the transaction histories of your orders. The transaction histories that are included in your CSV file include the captured payments data only. Authorization data is not included in your exported transaction histories.

To export your orders as a CSV file:

1. Sort and filter your orders.
2. From the **Orders** [🔗](#) page, click **Export**.
3. In the **Export orders** window:
 - a. Select the option for the orders that you want to export. For example, if you want to export your orders by date, then click **Export orders by date** and set the start and end dates for the orders that you want to export.
 - b. Choose the format that you want to export your orders to:

Exporting.

Header	Definition
Name	The order number as it appears in your store <u>admin</u>
Phone	The customer's phone number
Email	The customer's email address
Financial Status	Whether the order has been paid, authorized, refunded, and so on.
Paid at	The date when the payment was captured for an order.
Fulfillment Status	Whether the order has been fulfilled or is still pending.
Fulfilled at	The date when the fulfillment was completed for an order.
Accepts Marketing	Whether this customer has agreed to accept marketing from your store.
Currency	The currency that the order was paid in.
Subtotal	The order's subtotal before shipping and taxes.
Shipping	The total cost of shipping for the order.
Taxes	The amount of taxes charged on an order.
Total	The total cost of the order.
Discount Code	The discount code that was applied to the order.

Id.

62. Plaintiff has been damaged by Defendant's infringement of the '698 Patent.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff respectfully requests the Court enter judgment against

Defendant:

1. declaring that the Defendant has infringed the '087 Patent;
2. awarding Plaintiff its damages suffered as a result of Defendant's infringement of the '087 Patent;
3. declaring that the Defendant has infringed the '989 Patent;
4. awarding Plaintiff its damages suffered as a result of Defendant's infringement of the '989 Patent;
5. declaring that the Defendant has infringed the '356 Patent;
6. awarding Plaintiff its damages suffered as a result of Defendant's infringement of the '356 Patent;
7. declaring that the Defendant has infringed the '698 Patent;
8. awarding Plaintiff its damages suffered as a result of Defendant's infringement of the '698 Patent;
9. awarding Plaintiff its costs, attorneys' fees, expenses, and interest; and
10. granting Plaintiff such further relief as the Court finds appropriate.

JURY DEMAND

Plaintiff demands trial by jury, Under Fed. R. Civ. P. 38.

Dated: July 20, 2018

Respectfully Submitted

/s/ Raymond W. Mort, III

Raymond W. Mort, III

Texas State Bar No. 00791308

raymort@austinlaw.com

THE MORT LAW FIRM, PLLC

106 E. Sixth Street, Suite 900

Austin, Texas 78701

Tel/Fax: (512) 865-7950

ATTORNEYS FOR PLAINTIFF